Q-Commerce(Foodtuck) Marketplace Technical Foundation

1. System Architecture Overview

Diagram

Components and Roles

- **Frontend**:
  - Built with Next.js and Tailwind CSS for a responsive, user-friendly UI.
  - Handles user interactions, renders data from APIs, and manages client-side state.
- **Sanity CMS**:
  - Manages content for products, categories, and user data.
  - Provides a schema-driven content structure for dynamic updates.
- **APIs**:
  - Integrates third-party services for inventory, shipping, and real-time tracking.
  - Facilitates communication between the frontend and backend systems.
- **Database**:
  - Stores user, order, and product details.
  - Optimized for quick read/write operations to ensure smooth user experiences.

---

2. Key Workflows

User Workflows

1. Adding Products to Cart

1. **User Action**: User selects a product and clicks "Add to Cart."
2. **Frontend**:
   - Sends a POST request to the `/cart` endpoint.
   - Updates local cart state for immediate feedback.
3. **Backend**:
   - Validates product availability via the `/products/{id}` endpoint.
   - Updates the cart in the database.

4. **Response**:
    - Returns updated cart details to the frontend for rendering.

2. Checkout Process

1. **User Action**: User clicks "Proceed to Checkout."
2. **Frontend**:
    - Validates user inputs and cart data.
    - Sends a POST request to `/checkout` with order details.
3. **Backend**:
    - Processes payment via a third-party payment API.
    - Creates an order entry in the database.
    - Sends confirmation email to the user.
4. **Response**:
    - Returns order confirmation details to the frontend.

---

3. Category-Specific Instructions

Q-Commerce Features

- **Real-Time Inventory Updates**:
    - Endpoint: `/inventory/{productId}`
    - Method: GET
    - Purpose: Fetches current stock levels for a product.
- **Delivery SLA Tracking**:
    - Endpoint: `/delivery-status/{orderId}`
    - Method: GET
    - Purpose: Provides real-time delivery status for an order.
- **Express Delivery Workflows**:
    - Endpoint: `/express-delivery-status`
    - Method: GET
    - Purpose: Fetches SLA and tracking updates for express deliveries.

---

4. API Specifications

Endpoint	Method	Purpose	Request Body	Response Example
`/products`	GET	Fetches all product details	N/A	`{ "id": 1, "name": "Product A", "price": 100 }`
`/products/{id}`	GET	Fetches details of a single product	N/A	`{ "id": 1, "name": "Product A", "price": 100, "stock": 50 }`
`/cart`	POST	Adds product to cart	`{ "productId": 1 }`	`{ "cartId": 123, "items": [...] }`
`/checkout`	POST	Processes order and payment	`{ "orderDetails": {...} }`	`{ "orderId": 456, "status": "success" }`

---

5. Data Schema Design

Sanity Schema: Product

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'category', type: 'reference', to: [{ type: 'category' }], title: 'Category' }
  ]
};
```

Database Schema

- **Users**:
  - Fields: `id`, `name`, `email`, `password`, `address`
- **Products**:
  - Fields: `id`, `name`, `price`, `stock`, `category`
- **Orders**:
  - Fields: `id`, `userId`, `productIds`, `totalAmount`, `status`

---

6. Technical Roadmap

Milestones and Deliverables

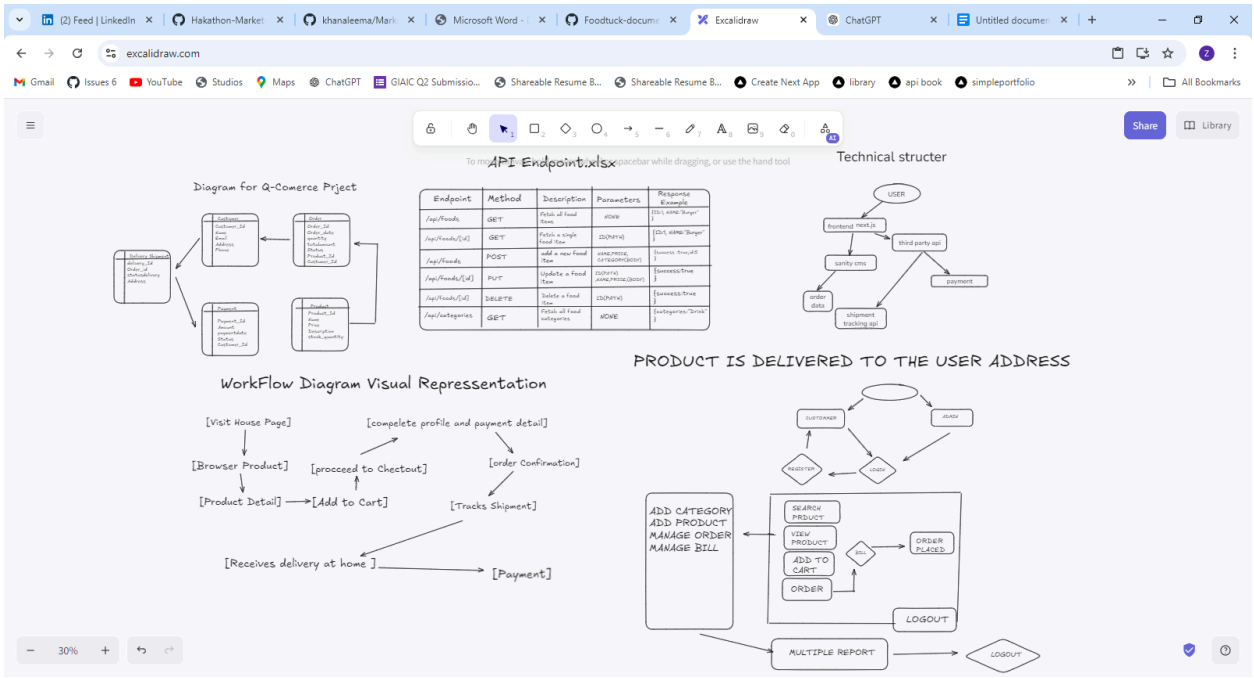
1. **Milestone 1**: Setup Environment (Week 1)
   - Configure Next.js, Tailwind CSS, and Sanity CMS.
   - Implement basic routing and layout.
2. **Milestone 2**: Core Features (Week 2-3)
   - Develop product catalog and cart functionality.
   - Integrate Sanity CMS.
3. **Milestone 3**: API Integrations (Week 4)
   - Add real-time inventory and delivery tracking endpoints.
4. **Milestone 4**: Testing and Deployment (Week 5)
   - Perform end-to-end testing.
   - Deploy to Vercel or another hosting platform.

---

7. Collaboration Tools

- **Communication**: Slack, Discord, Google Meet
- **Version Control**: GitHub
- **Project Management**: Trello, Jira

Diagrams:-


excalidraw.com
Share
Library
API Endpoint.xlsx
Diagram for Q-Comerce Prject
Technical structer
USER
frontend next.js
third party api
sanity cms
payment
order data
shipment tracking api
WorkFlow Diagram Visual Repressentation
[Visit House Page]
[Browser Product]
[Product Detail]
[Add to Cart]
[proceeed to Cheetout]
[compelete profile and payment detail]
[order Confirmation]
[Tracks Shipment]
[Receives delivery at home]
[Payment]
PRODUCT IS DELIVERED TO THE USER ADDRESS
CUSTOMER
ADMIN
REGISTER
LOGIN
ADD CATEGORY
ADD PRODUCT
MANAGE ORDER
MANAGE BILL
SEARCH PRODUCT
VIEW PRODUCT
ADD TO CART
ORDER
BILL
ORDER PLACED
LOGOUT
MULTIPLE REPORT
LOGOUT
30%